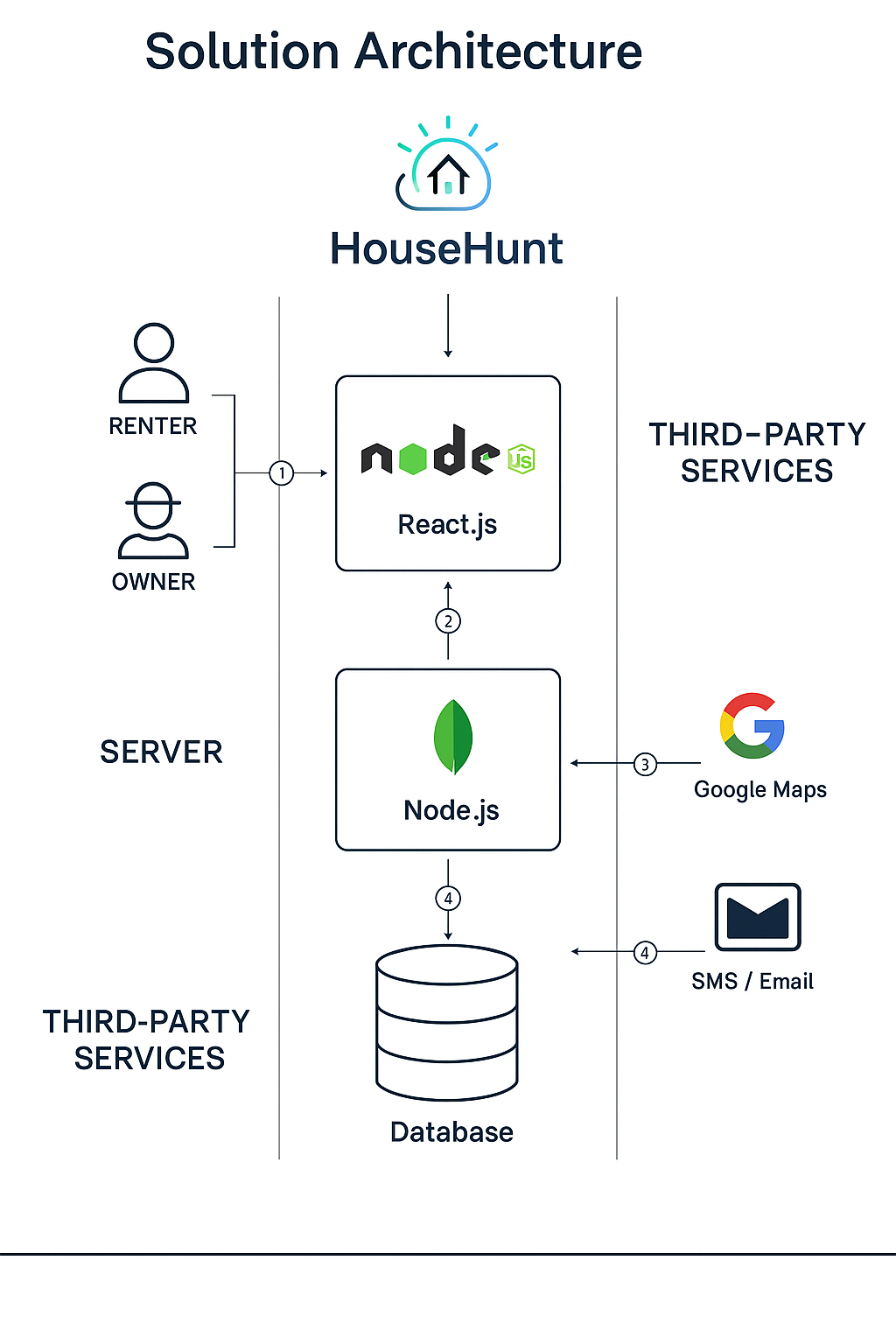
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 21/06/25 |
| Team ID | LTVIP2025TMID58222 |
| Project Name | House-Hunt: Finding-Your-Perfect-Rental-Home |
| Maximum Marks | 4marks |

**Technical Architecture:**

****

**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web-based UI for renters, owners, and admins to interact with the system. | React.js, Tailwind CSS |
|  | Application Logic-1 | Handles user authentication and role management (Renter/Owner/Admin). | Node.js, Express.js |
|  | Application Logic-2 | Business logic for property listing, search, and filtering. | Node.js, Express.js |
|  | Application Logic-3 | Booking management, messaging, review system, and wish list functionalities. | Node.js, Express.js |
|  | Database | Stores user info, property details, bookings, messages, and reviews. | MongoDB |
|  | Cloud Database | |  | | --- | |  |  |  | | --- | | Cloud-hosted database service for real-time access and scalability. | | MongoDB Atlas |
|  | File Storage | Stores images and documents related to rental listings. | Cloud, Firebase Storage |
|  | External API-1 | For Google Maps integration to show property location and directions. | Google Maps API |
|  | External API-2 | For social login (Gmail / LinkedIn) and email verification. | Google OAuth, LinkedIn API, SMTP or Firebase Auth |
|  | Machine Learning Model | Recommends properties based on user preferences and search history. | Python, Scikit-learn or TensorFlow (Optional) |
|  | Infrastructure (Server / Cloud) | Backend hosted on cloud platform for performance and availability. | Render, Vercel , or AWS EC2 |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Entire project is built on widely adopted open-source tools and libraries. | React.js, Node.js, MongoDB |
|  | Security Implementations | OAuth login, JWT tokens, HTTPS, input validation, and database sanitization. | Firebase Auth, JWT, Helmet.js |
|  | Scalable Architecture | Microservices-ready backend, cloud-hosted DB, and load balancing options. | Node.js with Express, MongoDB Atlas |
|  | Availability | Deployed on cloud platforms with high uptime and monitoring. | Render, Vercel, AWS |
|  | Performance | Fast response with server-side pagination, caching, and optimized DB queries. | Redis (optional), Express.js |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)